Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-36. (Canceled)

37. (Currently Amended) A computer implemented method comprising:

receiving, using a processor, a command to store a file in a file system having an inode, wherein the inode is usable to store metadata associated with the file;

responsive to the file having a size that is greater than a block size of blocks in the file system, storing a plurality of data of the file only in an integer number of blocks, wherein a first remainder data of the file remains after storing, and wherein the first remainder data is less than the block size; [[and]]

placing the first remainder data directly into the inode; and [[.]]

responsive to a second remainder data of the file still remaining after the first remainder data is placed into the inode, placing the second remainder data into an unused space of a partially used block of the file system, wherein the partially used block also stores data of another file.

38. (Currently Amended) The computer implemented method of claim 37 wherein placing the first remainder data directly into the inode further comprises placing the first remainder data in an extension area of the inode~~, wherein the extension area was~~ formerly reserved for the metadata, and the computer implemented method further comprises:[[.]]

updating a mode field in the inode to designate that the first remainder data of the file has been stored in the inode;

prior to performing the step of placing the second remainder data into an unused space of the partially used block of the file system, determining whether the partially used block exists and whether the partially used block has a sufficient free space to store the second remainder data;

updating the mode field in the inode to designate that the second remainder data has been stored in the partially used block; and

responsive to the partially used block becoming full from storing the second remainder data in the partially used block, removing the partially used block from a list of free shared blocks, wherein, for each free shared block on the list of free shared blocks, the list of free shared blocks contains a block number, a free byte quantity, and a pointer to a next free shared block.

39. (Canceled)

40. (Currently Amended) The computer implemented method of claim 37 further comprising:

~~wherein~~ when the size is less than the block size, ~~and wherein the method further comprises:~~ placing data of the file directly into the inode.

41. (Currently Amended) The computer implemented method of claim 40, further comprising:

~~wherein~~ when [[a]] the second remainder data remains after placing the data into the inode, ~~and wherein the computer implemented method further comprises:~~ placing the second remainder data into an unused space of a first block of the file system, wherein the first block also stores data of another file, and wherein the first block comprises a last block of the another file.

42-43. (Canceled)

44. (Currently Amended) A recordable-type computer readable computer readable medium on which is stored a computer program product executable in a data processing system, the computer program product comprising:

instructions for receiving a command to store a file in a file system having an inode, wherein the inode is usable to store metadata associated with the file;

instructions for, responsive to the file having a size that is greater than a block size of blocks in the file system, storing a plurality of data of the file only in an integer number of blocks, wherein a first remainder data of the file remains after storing, and wherein the first

remainder data is less than the block size; [[and]]

instructions for placing the first remainder data directly into the inode; and[[.]]

instructions for placing a second remainder data into an unused space of a partially used block of the file system responsive to the second remainder data of the file still remaining after the first remainder data is placed into the inode, wherein the partially used block also stores data of another file.

45. (Currently Amended) The recordable-type computer readable medium of claim 44 wherein placing the first remainder data directly into the inode further comprises placing the first remainder data in an extension area of the inode formerly reserved for the metadata, and the recordable-type computer readable medium further comprises:

instructions for updating a mode field in the inode to designate that the first remainder data of the file has been stored in the inode;

instructions for, prior to performing the step of placing the second remainder data into an unused space of the partially used block of the file system, determining whether the partially used block exists and whether the partially used block has a sufficient free space to store the second remainder data;

instructions for updating the mode field in the inode to designate that the second remainder data of the file has been stored in the partially used block; and

instructions for removing the partially used block from a list of free shared blocks responsive to the partially used block becoming full from storing the second remainder data in the partially used block, wherein, for each free shared block on the list of free shared blocks, the list of free shared blocks contains a block number, a free byte quantity, and a pointer to a next free shared block. ~~wherein the computer program product further comprises:~~

~~instructions for, responsive to second remainder data of the file still remaining after the remainder data is placed into the inode, placing the second remainder data into an unused space of a first block of the file system, wherein the first block also stores data of another file, and wherein the first block comprises a last block of the another file.~~

46. (Currently Amended) The recordable-type computer readable medium of claim 44, further comprising:

~~wherein the size is less than the block size, and wherein the computer program product further comprises:~~ instructions for placing data of the file directly into the inode when the size is less than the block size.

47. (New) A data processing system comprising:

a bus system;

a communications unit connected to the bus system;

a memory connected to the bus system, wherein the memory includes a set of instructions; and

a processing unit connected to the bus system, wherein the processing unit executes the set of instructions comprising:

instructions to receive a command to store a file in a file system having an inode, wherein the inode is usable to store metadata associated with the file;

instructions to store data of the file only in an integer number of blocks responsive to the file having a size that is greater than a block size of blocks in the file system, wherein a first remainder data of the file remains after storing, and wherein the first remainder data is less than the block size;

instructions to place the first remainder data directly into the inode; and

instructions to place a second remainder data into an unused space of a partially used block of the file system responsive to the second remainder data of the file still remaining after the first remainder data is placed into the inode, wherein the partially used block also stores data of another file.

48. (New) The data processing system of claim 47, wherein the instructions to place the first remainder data directly into the inode further comprise instructions to place the first remainder data in an extension area of the inode formerly reserved for the metadata, and the data processing system further comprises:

instructions to update a mode field in the inode to designate that the first remainder data of the file has been stored in the inode;

instructions to determine whether the partially used block exists and whether the partially used block has a sufficient free space to store the second remainder data prior to performing the step of placing the second remainder data into an unused space of the partially used block of the file system;

instructions to update the mode field in the inode to designate that the second remainder data has been stored in the partially used block; and

instructions to remove the partially used block from a list of free shared blocks responsive to the partially used block becoming full from storing the second remainder data in the partially used block, wherein, for each free shared block on the list of free shared blocks, the list of free shared blocks contains a block number, a free byte quantity, and a pointer to a next free shared block.